Enhancing Open-Set Recognition using Clustering-based Extreme Value Machine (C-EVM)

Henrydoss, James ; Cruz, Steve ; Li, Chunchun ; Günther, Manuel ; Boult, Terrance E

Abstract: In real-world deployments, machine learning applications find challenges when accessing ever-increasing volumes of data – the real world is open and often presents data from classes not seen in training. Open-set recognition is a growing area of machine learning addressing such problems. This research work advances the state-of-the-art in open-set recognition, the Extreme Value Machine (EVM), with a novel clustering-based extension (C-EVM) during training to improve the end-to-end prediction performance. The C-EVM combines Density-based spatial clustering of applications with noise (DBSCAN)-based clustering with a novel Nearby Clusters (NC) algorithm during model fitting to reduce computation while improving accuracy. Our experiments show a statistically significant improvement of 5-10% in macro F1-score over the state-of-the-art EVM on open-set testing using the KDD CUP-99 data set. Past work on open-set recognition often traded improved open-set robustness for a decrease in closed-set accuracy, whereas C-EVM outperforms the EVM in both closed-set and open-set recognition. Testing on subsets of ImageNet-2012 with varying numbers of classes, the C-EVM statistically significantly outperforms EVM when using deep features. A parameterless Hierarchical DBSCAN (HDBSCAN)-based C-EVM variant is introduced as part of this work that scales well for large data sets. Finally, both EVM and C-EVM can operate as kernel-free incremental learners, enabling these open-set multi-class classifiers to be useful for streaming and big data applications.

DOI: https://doi.org/10.1109/BigData50022.2020.9378012

Enhancing Open-Set Recognition using Clustering-based Extreme Value Machine (C-EVM)

James Henrydoss,[1] Steve Cruz,[1] Chunchun Li,[1] Manuel Günther,[2] Terrance E. Boult[1]

[1] Vision and Security Technology Lab
University of Colorado Colorado Springs
{jhenrydo, scruz, cli, tboult}@vast.uccs.edu

[2] Department of Informatics
University of Zürich
guenther@ifi.uzh.ch

***Abstract*—In real-world deployments, machine learning applications find challenges when accessing ever-increasing volumes of data – the real world is open and often presents data from classes not seen in training. Open-set recognition is a growing area of machine learning addressing such problems. This research work advances the state-of-the-art in open-set recognition, the Extreme Value Machine (EVM), with a novel clustering-based extension (C-EVM) during training to improve the end-to-end prediction performance. The C-EVM combines Density-based spatial clustering of applications with noise (DBSCAN)-based clustering with a novel Nearby Clusters (NC) algorithm during model fitting to reduce computation while improving accuracy. Our experiments show a statistically significant improvement of 5-10% in macro F1-score over the state-of-the-art EVM on open-set testing using the KDD CUP-99 data set. Past work on open-set recognition often traded improved open-set robustness for a decrease in closed-set accuracy, whereas C-EVM outperforms the EVM in both closed-set and open-set recognition. Testing on subsets of ImageNet-2012 with varying numbers of classes, the C-EVM statistically significantly outperforms EVM when using deep features. A parameterless Hierarchical DBSCAN (HDBSCAN)-based C-EVM variant is introduced as part of this work that scales well for large data sets. Finally, both EVM and C-EVM can operate as kernel-free incremental learners, enabling these open-set multi-class classifiers to be useful for streaming and big data applications.**

I. Introduction

Machine learning is critical in knowledge discovery and data mining and becoming widely used in dozens of different domains. While the vast majority of machine learning algorithms are designed with the assumption that training classes are all known, the real world will often present classes not seen during training. Scheirer *et al.* [30] have formalized the problem of open-set recognition where algorithms properly balance the risk of unknown classes with accuracy for known classes by rejecting inputs as being unknown. The recently introduced Extreme Value Machine (EVM) [29] is the current state-of-the-art for open-set multi-class recognition using hand-tuned or pre-trained features. In this paper, we advance the state-of-the-art in open-set classifiers with a clustering-based extension to the EVM (C-EVM), improving both speed and accuracy.

This open-set nature of problems has been known but ignored in the KDD community for a long time [16], [28], until recentl research efforts started to address it explicitly [14]. Open-set recognition can be viewed as a combination of multi-class recognition with novelty detection. Unlike work that treats it as a two-stage problem, novel-detection followed by multi-class recognition, open-set recognition combines the two into a single solution. This type of problem occurs naturally in data stream mining and evolving streams [11], [4], which also require incremental learning. While this paper does not explore incremental learning, EVM and C-EVM have natural incremental extensions.

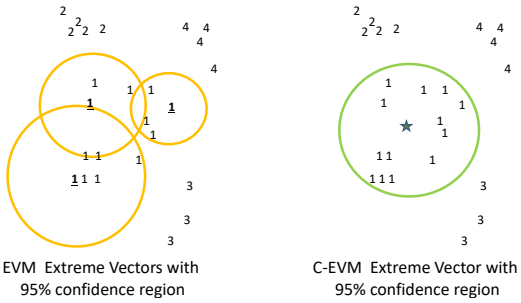

Fig. 1: EXTREME VECTORS IN EVM AND C-EVM. *Each number represents a point in a class, and the colored circles represent the 95% confidence interval around the Extreme Vectors (EVs). Standard EVM selects EVs that cover the data, but often take over more of the open-space and need more EVs, which are shown as 1, three in total. C-EVM uses cluster centroids, the star, as the location of EV points. Less EVs are used and often cover less in open space, thereby reducing open-set risk and improving open-set recognition performance when unknown classes are present.*

The novelty of this paper is based on our key insight, which combines the advantages of the EVM with prior data mining techniques that improve classification in other models by employing clustering-based data grouping or centroid coordinates during training [39], [12], [21], [6], [33]. The core theory behind the original EVM [29] is its Margin Distribution (MD) theorem, which provides a well-grounded theory for the probability of any point in space beyond the class margin, i.e., an outlier for the given class. The authors show that the Probability of Sample Inclusion (PSI) is given by a radial Weibull-based model. The EVM is an instance-based classifier, which uses a set-cover solution to choose which of the instances, i.e. the Extreme Vectors (EV), to keep as the basis of the probability model. While the Margin Distribution theorem and the PSI model are well-founded, Rudd *et al.* [29] assume, without any supporting evidence, that each PSI model should be centered on training sample points and only consider such points as extreme vectors. This paper challenges that assumption by hypothesizing that centroids of clusters will reduce the number of extreme vectors needed

and provide cleaner and more centered extreme vectors and, therewith, cause a more accurate model in open-set problems. The averaging to obtain centroids reduces noise and also allows, on average, better coverage with spherical Weibull-based probability models. Improving the coverage reduces the number of required models and improves generalizability with lower open-space risk. Using a clustering-enabled approach, we develop a novel classifier model, the *Clustering-based EVM (C-EVM)*, that improves accuracy and provides a more scalable algorithm. While there exists a plethora of clustering approaches, the majority of this paper uses a C-EVM model built using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) while selecting centroids for building PSI models. We also evaluate an hierarchical variant (HDBSCAN) inside of the C-EVM.

The main contributions include:

- Development of the novel C-EVM from Weibull-based models of cluster centroids.
- Evaluation of the C-EVM model on an open-set variation of the KDD CUP-99 intrusion detection data set and on a closed-set protocol for ImageNet-2012 showing statistically significant improvements in macro F1 measure over the state-of-the-art EVM.
- Ablation studies on KDD showing the improvements due to improved vector ratio from a lower number of extreme vectors, but not because of outlier rejection.
- Extension of the C-EVM to use HDBSCAN with automated parameter selection with similar performance on KDD CUP-99 data, showing that the gains are not from parameter tuning.
- Showing C-EVM improvements on both closed-set and open-set accuracy with a significant speedup.

II. Related Work

Scheirer *et al.* [30] found the open-set-based recognition method for computer vision problems and derived formal definitions for open-space and open-space risk. Günther *et al.* [15] extended these definitions for implementing open-set face identification and formulating the *open-set identification protocol* that outlines the procedures to deal with the unknowns received during query time. The primary objective of an open-set identification method is to correctly identify probe subjects that are present in the gallery while rejecting all others as unknown. Jain *et al.* [19] developed multi-class open-set classification using Pi-SVM and define the known and unknown class. Rudd *et al.* [29] presented the Extreme Value Machine (EVM) for computer vision, a novel multi-class classifier that can support incremental learning and image recognition based on the open-set concept. There have been other advances and application papers in open-set recognition, which can be found in a broader survey [5].

In this work, we extend the EVM and introduce a clustering-enabled EVM (C-EVM) using DBSCAN and HDBSCAN density clustering methods to enhance performance. As indicated in the literature [41], [13], [31], DBSCAN clustering can be an outstanding choice [40] as it can be used to process real-time data [13] and handle large data sets [41] with minimum input configuration parameters to yield a better prediction performance. Many previous research efforts confirm the benefit of outlier removal for yielding better performance [8], [38], [27], [24]. Elik *et al.* [7] proposed the use of DBSCAN to build anomaly detection for the removal of outliers. Chakrborty *et al.* [9], [8] showed performance improvements using incremental DBSCAN and incremental K-Means. Kumar *et al.* [22] implemented a fast DBSCAN clustering algorithm that enhance the performance of DBSCAN and is scalable for high-dimensional data sets on benchmark data sets with a speed improvement of 1.2 - 2 times.

Clustering data is an unsupervised machine learning technique that is widely practiced in the area of data mining with a wide range of applications. Ali *et al.* [2] employed K-Means-based approach to improve the accuracy of a decision-tree response classification task, reporting an increase of over 15% in classification accuracy. Theodorakis *et al.* [36] used a hierarchical clustering scheme to enhance classification accuracy and improve classification accuracy from 2.32% to 7.25% over COBWEB, an unsupervised conceptual clustering algorithm.

Deng *et al.* [12] implemented an efficient kNN classification with improved accuracy using a clustering-based approach. They employed K-Means to separate the whole data set into several parts, each of which undergoes kNN classification, yielding improved accuracy and efficiency on medical imaging data. Caron *et al.* [6] developed a novel, K-Means-based clustering approach for implementing large scale end-to-end training of convolutional networks that achieve significantly higher prediction performance than any previously published unsupervised learning method on transfer learning tasks.

III. Approach

Our work combines ideas from three major areas, open-set recognition, the EVM, and clustering.

A. Open-Set Recognition

Real-world classification tasks are limited by various factors. For example, when training a classifier or recognizer it is usually difficult to collect training samples for all classes that the classifier will see during deployment. Open-Set Recognition (OSR) provides a more realistic scenario where incomplete knowledge of the world is present during training and samples of unknown classes are seen during testing [30]. Jain *et al.* [19] define the known and unknown class categories as follows:

- *Known Known Classes*: Known classes are classes with distinctly labeled positive training examples. A known training example is a positive sample for a class of interest C_{i} to be classified, and also serves as a negative for other known known classes. Often, positive samples have corresponding semantic/attribute information.
- *Known Unknown Classes*: Known unknown classes are composed of known but uninteresting samples. These samples serve as negatives for all the known classes,

while the exact labels of known unknown samples are usually disregarded.

- *Unknown Unknown Classes*: These classes are totally unknown to the classifier, i.e., no samples of these classes are seen during training, and usually no side-information such as the semantics or attributes is available.

Scheirer *et al.* [30] formalize the open-set recognition method. Consider an example with a large ball S_O, consisting of both the open-space O and all of the positive training examples. The recognition function is f where $f(x) = 0$ when the class of interest, y, is not fully recognized, and $f(x) = 1$ when it is fully recognized. Under these conditions, the open-space risk of $R_O(f)$ can be defined as follows:

$$R_O(f) = \frac{\int_O f(x)dx}{\int_{S_O} f(x)dx} \tag{1}$$

where open-space risk is considered to be the fraction in terms of Lebesgue measure of positively labeled open-space compared to the overall measure of positively labeled space that includes the space near positive examples. Scheirer *et al.* [30] formally defined the open-set recognition as follows. **Open-Set Recognition Problem Definition:** Let samples $V = \{v_1, \ldots, v_m\}$ from P be our positive training set of data and samples $K = \{k_1, \ldots, k_n\}$ from other known classes K be our negative training data samples. Let U be the larger universe of allowed unknown (negative) classes which appear only during testing. Let $T = \{t_1, \ldots, t\_z\}$ with $t_i \in P \cup K \cup U$ be our test data, where the openness problem is > 0. Given the training data $V \cup K$, an open-space risk function R_O, and an empirical risk function R_ϵ, open-set recognition is to find a measurable recognition function $f \in H$, where $f(x) > 0$ implies positive recognition, and f is defined by minimizing the *open-set risk*:

$$R_O(f(U)) + \lambda_r R_\epsilon(f(V \cup K)) \tag{2}$$

where λ_r is a regularization constant. This equation defines the open-set recognition as minimizing the open-set recognition risk by combining open-set risk and empirical risk, which is related to the measurement performance in the allocated space of recognition function operation. Given conditions of assumptions about the function $f \in H$, this definition balances what is known via $(V \cup K)$ and the open-space risk R_O in association with the unknown classes U. According to Scheirer *et al.* [30], the *degree of openness* can be formalized by considering the number of classes used in training and the number of classes seen in testing.

B. Extreme Value Machine (EVM)

In the original formulation by Rudd *et al.* [29], the Extreme Value Machine (EVM) is a classifier designed for open-set recognition. In a given feature space, the EVM represents classes by sets of extreme vectors, each of which is accompanied by a certain Probability of Sample Inclusion (PSI) model. For a given test sample, the PSI is computed for each extreme vector. If any probability exceeds a specified threshold, it is assigned to the corresponding class of the extreme vector; otherwise, the sample is declared to be unknown.

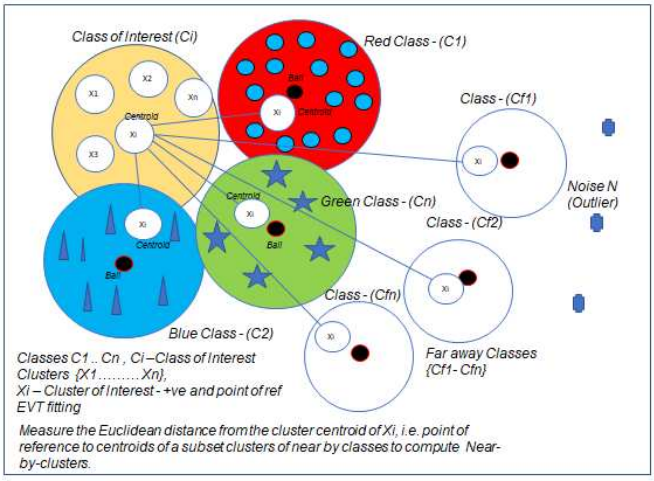

Fig. 2: NEARBY CLUSTERS. *To estimate the PSI model for the centroid of the cluster in the current class of interest (yellow), only samples from nearby clusters of other classes (red, green, blue) are utilized while faraway clusters (white) are disregarded.*

To build a PSI model in the EVM, each of the N training points is compared to other training points of a different class by a certain distance function, which results in time complexity of $O(N^2M)$ where M is the dimensionality of the feature space. To select the data to be used for PSI model fitting, the distances need to be sorted, which requires $O(N^2 \log N)$ operations and only the smallest τ distances to any other known class are kept. Then, the number of PSI models are reduced by a greedy set-cover algorithm, keeping the resulting exemplars as extreme vectors. For this process, the inclusion probability of samples in the same class are computed, and many training samples (including their expensively computed PSI models) are thrown away.

The EVM model utilizes the Margin Distribution (MD) of each sample point with reference to its closest negative samples. EVM suffers from the point-by-point computation of distances for building the MD that requires the τ closest samples from any negative class and often ends up with many extreme vectors, which cover more open-space than ideally desired. To enhance accuracy and reduce cost, we introduce a clustering-based approach during training in the next sections.

C. Extreme Vector Selection

The key idea of using clustering is to remove the necessity to compare each of the N features to each other. Instead, we perform clustering only per class, which reduces the number of comparisons dramatically to $O(D^2M)$ where $D \ll N$ is the number of samples of the largest class. Once all clusters for all classes are obtained, we compute the centroid of each cluster and treat it as an extreme vector of the C-EVM. The key parameter of DBSCAN is the maximum distance ϵ allowed between two samples to be considered as part of the same cluster. Outliers, i.e., samples with distances greater than ϵ to all other sample are removed from further consideration.

The clusters' centroids are used as the point of reference for the C-EVM fitting, which is depicted in Fig. 2. For each cluster

of each class, we find the Nearby Clusters (NC) by computing and sorting the distance from its centroid to all other centroids of other classes clusters. To arrive at a decision, which clusters to consider as NC, we select the number of clusters β to include. Then, we start with the cluster containing the largest number of points and declare its centroid as an extreme vector. Clusters that fail to be a part of the NC due to having a larger distance than β are excluded from the computation.

To fit a PSI model [29], we need the τ closest points from other classes, which we collect from the clusters obtained via NC. In order to decide which points are the closest, we compute distances to all points in a cluster, for which we use all points of nearby clusters as negatives. After all PSI models are fit, we perform Cluster Covering (CC), whereby we determine which other centroids from that class are covered, i.e., have a probability greater than a fixed threshold to be included by another EVT model of the same class. We select the cluster with an uncovered centroid that have the most points in that class as the extreme vectors.

D. HDBSCAN

One of the major challenges in using DBSCAN clustering for large data sets is selecting the right configuration parameters `MinPts` and ϵ manually. We want the C-EVM to be used for large data sets and so we extend the EVM design to HDBSCAN that has no parameter tuning. HDBSCAN is an enhanced DBSCAN clustering method that performs DBSCAN over varying epsilon values and integrates the result to find a clustering that gives the best stability over epsilon [25], [26]. This enables HDBSCAN to find clusters of varying densities and is more robust to parameter selection. It means that the HDBSCAN enabled C-EVM can yield better clustering results with less or no parameter tuning while facilitating the use of C-EVM right away for large data sets. In addition, HDBSCAN provides outlier detection using GLOSH algorithm, a condensed cluster hierarchy, robust single linkage cluster hierarchy, and reach ability distance minimal spanning tree algorithm support [26].

IV. Experiments

To evaluate the open-set recognition performance of the C-EVM model, we compare it with the open-set multi-class classifier EVM [29]. The EVM is the current state-of-the-art for open-set multi-class recognition using hand-tuned or pre-trained features. It was tested on ImageNet and UCI features [29] and evaluated on intrusion detection using KDD data [18]. For ease of comparison, this paper will use the same data sets, and we will perform closed-set experiments on ImageNet. We use macro F1-measure as computed in Python's `SKlearn` library as the primary evaluation criterion, although we include accuracy for closed-set testing to compare with past work. For a given threshold γ, F1-measure is computed as:

$$\mathrm{F1}(\gamma) = \frac{2 \cdot \mathrm{Pr}(\gamma) \cdot \mathrm{Re}(\gamma)}{\mathrm{Pr}(\gamma) + \mathrm{Re}(\gamma)} \tag{3}$$

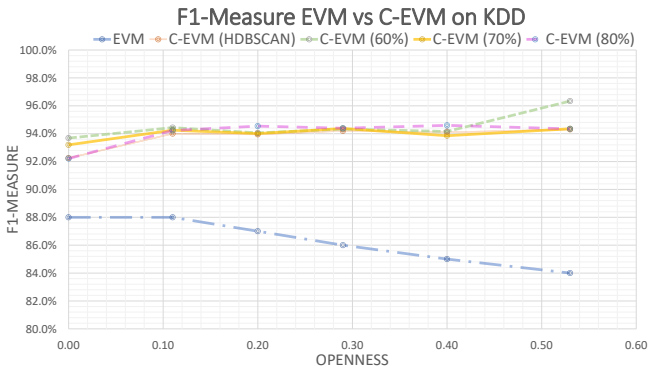

Fig. 3: EXPERIMENTS ON KDD. *Performance comparison on KDD data sets of C-EVM with previous state-of-the-art EVM at various levels of openness and various PSI thresholds.*

with:

$$\mathrm{Pr}(\gamma) = \frac{\mathrm{TP}(\gamma)}{\mathrm{TP}(\gamma) + \mathrm{FP}(\gamma)} \qquad \mathrm{Re}(\gamma) = \frac{\mathrm{TP}(\gamma)}{\mathrm{TP}(\gamma) + \mathrm{FN}(\gamma)} \tag{4}$$

A True Positive (TP) is an outcome where the model predicts the positive class as a positive class with a probability higher than γ. In KDD case, an intrusion will be classified into its correct attack category, a benign attempt as being benign, and an unknown sample as none of the known classes. A false positive sample is assigned to the wrong class with probability greater than γ, which includes that an unknown sample is classified as any of the known classes. A false negative occurs when a the probability of a certain known sample for its correct class is below threshold γ.

A. Open-Set Experiments on KDD Cup-99

Experiments include sample sets that consist of a set of known known and unknown unknown classes with varying openness levels for every run grouped as separate batches – unlike other work, we do not provide known unknown classes to train our algorithms. To change the open-set configuration, and to make it more open, we change the balance of known vs. unknown classes, reducing the known classes and associated training instances for each batch. As shown in Tab. I, using 14 classes for testing and varying the number used for training provides a range of openness.

We conduct open-set recognition experiments with the following model configuration parameters: $\epsilon = 0.3$, `MinPts` $= 1$ (both DBSCAN), 50% cover threshold, and we use PSI probability prediction thresholds for C-EVM of $\gamma = 60\%$, 70%, and 80%.

In Tab. I, we include all test scenario cases with varying openness levels performed using multiple batches for the C-EVM and the standard EVM. At any of the thresholds, using a two-sided paired t-test across the differences in openness, the C-EVM is statistically significantly better than the EVM with $p < 0.0001$. The data is graphically presented in Fig. 3, summarizing the test cases with varying openness levels performed using multiple batches for the C-EVM to EVM. When comparing the positive measures (not shown), C-EVM provides better precision and true positive predictions over

Train	Test	Openess	EVM	C-EVM (60%)	C-EVM (70%)	C-EVM (80%)	C-EVM (HDBSCAN)	C-EVM (K-Means)
14	14	0.00	88.0%	93.7%	93.2%	92.2%	92.3%	93.2%
11	14	0.11	88.0%	94.4%	94.2%	94.2%	94.0%	93.3%
9	14	0.20	87.0%	94.1%	94.0%	94.5%	94.0%	93.7%
7	14	0.29	86.0%	94.3%	94.4%	94.4%	94.2%	91.8%
5	14	0.40	85.0%	94.2%	93.9%	94.6%	94.1%	93.2%
3	14	0.53	84.0%	96.3%	94.3%	94.3%	94.3%	93.9%

TABLE I: EVM vs. C-EVM. *F1-measure based open-set recognition performance at various levels of openness on KDD. C-EVM using DBSCAN with different PSI thresholds and also using HDBSCAN and K-Means variants are shown. Three columns show varying PSI thresholds (60%, 70%, 80%) and it can be seen there is not much difference with variation in the threshold, or the use of HDBSCAN, K-Means (where the number of clusters set to 11) all of which are significantly better then the classic EVM. For C-EVM we used three levels 60%, 70%, 80% and for EVM we used 50%.*

the EVM. We also observe that, as the model becomes more open, the C-EVM maintains a constant F1-score performance while the EVM drops in performance. This shows that the C-EVM model correctly identifies the different attack types present in the KDD CUP-99 data set. The average performance across openness is best at 60% PSI threshold. As we increase the probability threshold to higher values, the average F1-score performance degrades slightly. However, there is not a statistically significant difference between the thresholds. We conclude that over a broad range of thresholds, the C-EVM model provides better performance when compared to the state-of-the-art.

1) Open-Set Extreme Vector Measurement

We now return to our hypothesis that the performance gains for C-EVM are related to the use of centroids, decreasing the number of extreme vectors needed. Each EV is associated with a radial inclusion function that is defined by the functional modeling using the PSI function. We use the EV to compute the Vector Ratio (VR) which is a standard measure of the model compactness and generalization ability. The VR computes how many of the training samples are kept in the model. According to Vapnik [37], VR is a scaled form of approximation of generalization error, and a smaller VR provides a better model generalization as well as the formation of compact models. We derive the VR for a model as the number of EVs by counting the number of EVs retained by the model divided by the total number of training samples. For KDD data set openness of 0%, 11% 20%, 29%, and 40%, the EVM yields vector ratios of 2.10, 2.07, 1.34, 1.33, and 1.95, respectively. For the same data, C-EVM has vector ratios of 0.87, 0.79, 0.37, 1.59, and 1.48 respectively. A two-tail paired T-test yields that the C-EVM is statistically significantly better ($p = 0.012$). Thus, we can conclude that C-EVM has a better VR providing better generalization, which supports our hypothesis that clustering is choosing points for the PSI models that provide better coverage with fewer points.

2) Ablation Study: C-EVM and Outliers

Outliers can play a major role in classifiers and in data mining. Syafrudin *et al.* [34] implemented a hybrid prediction model that used DBSCAN-based outlier detection and Random Forest classification. They removed outliers from the Internet of Things sensor data and provided a highly accurate fault detection system during the manufacturing process. Various other research efforts [1], [23] proposed an outlier-based model that enhanced the prediction performance. In an effort to understand the C-EVM improvement in prediction performance, an alternative hypothesis is that using the default DBSCAN will allow the system to ignore outliers that could impact PSI model fitting and overall performance. To assess this, we ran two variations on the KDD data, one that removes outliers and one that includes them.

The DBSCAN algorithm finds all points close to a given point, and if there are more than the number of neighbors defined by `MinPts`, it considers them as part of the same cluster as core points. If it cannot assign a point to any cluster, this point is defined as noise or an outlier to that class and discarded by the DBSCAN algorithm.

To measure the impact of these outliers during C-EVM fitting, we conduct a test with and without including the outliers as part of the C-EVM training. To perform this, the outliers are added or ignored by the Nearby-Cluster (NC) algorithm. Similar to our earlier experiments, we perform these experiments on various batches of the KDD data set using different openness levels. For KDD data set openness of 0%, 11%, 20%, 29%, and 53%, the C-EVM F1-score with outliers was 93.6, 94.4, 94.1, 94.0, and 96.3, respectively. Ignoring outliers, the F1-scores were 93.1, 93.9, 94.1, 94.2, and 94.4, respectively. With a two-sided paired T-test, we get a p-value of 0.22, and we cannot reject the null hypothesis that there is no difference between the scores with and without outliers. We hypothesize that this lack of difference is because the outliers are far enough away that they do not impact the closest negatives used for PSI model fitting.

3) C-EVM using HDBSCAN

Most of the open-set experiments conducted in this work use DBSCAN-enabled clustering implemented as part of the C-EVM design, where we used a fixed ϵ and `MinPts`. In an effort to evaluate the C-EVM model's performance with alternate clustering methods, we extended the C-EVM code base with HDBSCAN [25] clustering. Using the KDD CUP-99 data set, we ran C-EVM with HDBSCAN and show the results in Fig. 3 and Tab. I. We found no statistically significant difference ($p > 0.05$) with C-EVM that used DBSCAN with fixed parameters and any of the evaluated ϵ values. Thus we can conclude that it was not a magic parameter tuning that provided the advantage of C-EVM over EVM.

Openness	# Samples	C-EVM	EVM	Speedup
0.00	30495	3.34	6.89	2.06
0.11	27927	2.89	9.65	3.34
0.20	20489	3.00	10.69	3.56
0.29	16069	2.62	12.76	4.87
0.40	16071	2.37	9.07	3.82
0.53	13556	1.12	11.73	10.43

TABLE II: C-EVM VS EVM TRAINING TIME. *The training time is measured as milliseconds per processed point on the different KDD subsets.*

4) Speed comparison

One of the core objectives for C-EVM is speed improvement. Thus, we report the timing comparison for training on KDD data. All experiments are implemented in Python 2.7 and ran on an Intel i7 processor with 16GB of RAM.

If we presume the number of points per cluster to be constant, the overall asymptotic complexity is the same for C-EVM and EVM, but the constants can be very different. As we can see from Tab. II, C-EVM is more than three times faster than EVM on average, and for a smaller number of classes its speed advantage increases. As the number of classes increase, the average number of points per cluster decrease, thus reducing the speed advantage of using clusters over individual points. A two-sided paired T-test comparing the speed difference shows that it is statistically very significant with $p = 0.0001$.

B. Closed-set Comparison on KDD

In many prior open-set papers, there was an inherent trade-off, open-set robustness often came at the cost of close-set accuracy. This often occurred because, on a somewhat ambiguous input, an open-set algorithm is likely to reject it while a closed-set algorithm can still guess. With a small number of classes, or a binary problem, closed-set gets a strong advantage from guessing. Many machine learning researchers have devised mechanisms to perform optimal classifiers that provide better accuracy and detection performance over the KDD CUP-99 data set. In this section, we study and compare the performance of our open-set classifier models EVM and C-EVM on the KDD data set against the previously published closed-set machine learning results. To prepare a comparative study, we only use machine learning algorithms included in the results of Tavallaee *et al.* [35]. We use prediction accuracy as the primary technique to evaluate the above-mentioned classifiers with the open-set classifiers. Based on [35], the machine learning algorithms J48 and Decision Tree were the two best-performing methods with a prediction accuracy of 93.82% and 93.51%, respectively. So while EVM provided state-of-the-art in open-set testing using F1-measure in [18], we see EVM fell well off the mark with the closed-set accuracy of 90.01%. The novel C-EVM classifier provided state-of-the-art closed-set accuracy at 94.41% as well as statistically significantly better open-set performance.

C. Closed-set Comparison on ImageNet

In an effort to show the generality of the performance improvements of the C-EVM classifier, we evaluate it on a large data set using deep features. We use a subset of the popular computer vision data set, ImageNet, for evaluating the performance of the C-EVM algorithm. ImageNet is an image database organized according to the Word-Net hierarchy in which each node of the hierarchy is represented by thousands of images. In this case, we use the same deep features reported in [29], but since they did not report exactly what classes were used, or how they did their open-set testing, we do standard closed-set testing with a varying number of classes. We will release code to reproduce all our experiments.

Our hypothesis is that by implementing DBSCAN-based clustering, which forms meaningful clusters from the training data, and incorporating the proposed Near-by-Clusters during PSI model fitting, combined with a cluster-covering-based model reduction technique, we expect a significant enhancement in prediction performance, even in a closed-set evaluation. As we saw for KDD, a performance gain for C-EVM at openness 0 (closed-set) is expected to provide better performance at different levels of openness.

To test our hypothesis, we compare the C-EVM performance with the EVM algorithm on the same ImageNet data set. For all the experiments included as part of this testing, the DBSCAN configuration parameters are set up with $\epsilon = 0.3$ and $\texttt{MinPts} = 1$. We use the following test setups for our experiments: setup-I (5 classes, 6500 training images, 250 test samples), setup-II (20 classes, 26000 images, 1000 test samples), setup-III (50 classes, 65000 images, 2500 test samples), and setup-IV (100 classes, 130000 images, 5000 test samples).

In Fig. 4, we include a comparison of open-set prediction performances of C-EVM and EVM models using the ImageNet data set. We observe the C-EVM F1-measure increase in the range of 3.5% to 12% over the EVM's performance using the same test setup; the advantage increases with the number of classes. In conclusion, the C-EVM yields a better prediction performance than that of the original EVM when tested against the ImageNet data set. Based on the results, we observe that these two F1-measures are statistically significantly different based on the test runs with $p = 0.04$. This shows that our proposed design of C-EVM enhances the prediction performance of the existing EVM classifier with an average F1-measure improvement of 5.15%.

V. Discussion

In this section, we discuss the experimental results and their broader implications. We analyze the primary reason behind the performance improvement of C-EVM model recorded for our experiments on the KDD data set. In our design, to enhance the prediction performance of the current EVM model, we implement a DBSCAN-based clustering approach as part of our C-EVM model design during training time. In addition to clustering, the C-EVM design implements the following algorithms as part of the design: Near-by-Clusters (NC) and Cluster-Covering (CC) method during the Probability of Sample Inclusion PSI-based model generation. The NC algorithm uses only a small set of nearby negative

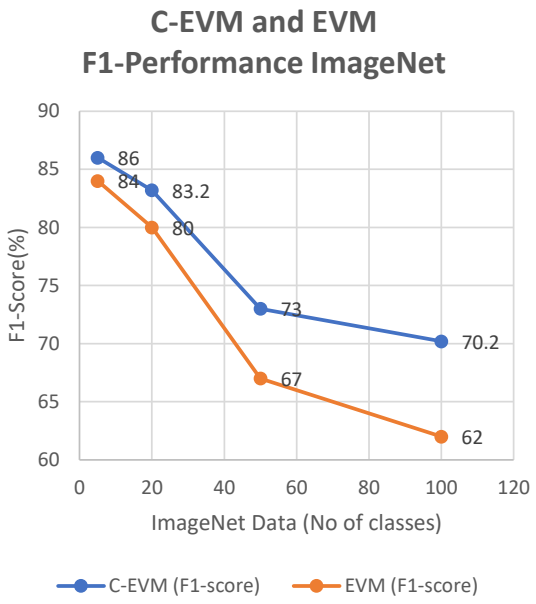

Fig. 4: C-EVM AND EVM OPEN-SET PERFORMANCE USING IMAGENET. *The number of classes is varied in a closed-set testing paradigm. As the number of classes increases, there is greater confusion, and both algorithms degrade, but the C-EVM is statistically significantly better ($p = 0.04$), and its advantage increases as the number of classes increases.*

cluster points for performing a PSI model fitting based on negative samples. The CC algorithm performs a Weibull PDF estimation of positive cluster points to measure whether they are already covered by other PSI models of the same class. The probability value of cluster coverage is high, the CC algorithm defines that point as already covered, so it skips them and does not perform any EVT negative fitting. Based on the results using the KDD CUP-99 data set and a subset of the ImageNet data set, we observe a reasonable increase in F1-measure of the C-EVM model in comparison to EVM. We have also conducted experiments using non-IDS data sets that include letter, MNIST-digits, shuttle and sat-image and observe significant F1-score performance improvement using the C-EVM [17]. We found that for a wide range of thresholds the C-EVM had stable performance and outperformed the EVM.

In alignment with earlier work that improves model performance using clustering-based approaches [10], [3], [20], [32], [6], our research work confirms that DBSCAN-based clustering during training helps to improve the prediction performance. Typically, for a given set of data points in a feature space, the clustering function groups relevant points that are closely packed together. We hypothesize that clustering during training improves the performance because it reduces the open-space risk and noise. In our implementation, a better clustering method, i.e., DBSCAN, which forms densely packed clusters and removes the outliers, and nearby clustering algorithms that group the samples according to the features which aligns with the right side of the class boundary. We showed that the gain was not likely due to outlier removal but from the averaging property of centroids over the cluster. Considering the algorithms underlying the results in Sec. IV-B, we note that each of the other tested algorithms built their classifiers out of raw examples.

Given that the only major difference between the final models for EVM and C-EVM is the use of centroid-based features rather than raw exemplars, it suggests that such features may be better suited and that extensions of other algorithms to use such features should be explored.

VI. Conclusion and Future Work

In this work, we improve open-set recognition performance for intrusion detection over the current state-of-the-art EVM algorithm. Also, we improve its training and operational speed. Building on the insights of past work using clustering [2], [36], [39], we recognize the representation and quality of data instances are vital factors that affect the classifier accuracy. We hypothesize that using cluster centroids in place of raw examples would improve the EVM model's accuracy and speed. We build our initial training using a clustering-based approach to yield quality data for our EVM classification. This proposed design enhances the prediction performance of the existing EVM model by implementing a DBSCAN-based clustering method combined with the Nearby Clustering (NC) and Cluster-Covering (CC) algorithms during Probability of Sample Inclusion (PSI) model fitting.

We successfully develop a new Clustering-based EVM (C-EVM) machine learning model that enhances accuracy and prediction performance. We observe statistically significant improvement in prediction performance, with an F1-measure increase in the order of 5 to 10 percent on open-set protocols on the KDD CUP-99 data set. Also, we saw significant improvements in speed and some improvement in closed-set accuracy over prior work. Thus, we conclude that C-EVM is the new state-of-the-art classifier on the KDD data set with strong open-set performance. While our objective was open-set performance, we found that the C-EVM improved performance overall, including closed-set performance. We showed it advanced closed-set accuracy on standard KDD testing. We have also verified the F1-measure performance improvement using closed-set testing by using deep-feature vision data set ImageNet-2012. Thus, we see the improvement over EVM is not feature representation specific.

In a nutshell, C-EVM development advances the state-of-the-art open-set classification performance by using the DBSCAN-based clustering approach. We have also extended the design to HDBSCAN-based clustering to verify the C-EVM model's performance with other clustering schemes and showed it was not because of hand-selected parameters.

This research effort paves the way for a new direction to build an improved, unsupervised, and incremental open-set machine learning model. Our underlying insight that clustering will improve performance is not new as variations on it have appeared from time to time in literature. The novelty of C-EVM was integrating clustering with EVM at multiple levels

to improve open-set performance. However, we expect that the Knowledge Discovery and Data Mining application field would benefit from exploiting this insight. While we built the C-EVM with clustering at multiple levels, it is often straightforward to use clustering and centroids to reduce the initial data set and get a performance boost.

While this paper did not experimentally explore it, the C-EVM can easily add new classes, as each new class just finds the nearest other class, and only a few already trained classes would need to be updated. Future work should report on the performance gains from such incremental usage and our public release of code will allow others to do that for their incremental problems.[1]

Acknowledgment

This research is based upon work supported in part by NSF IIS-1320956 and in part by the Office of the Director of National Intelligence (ODNI) via IARPA R&D Contract No. 2014-14071600012 and in part by DARPA via SAIL-ON Contract # HR001120C0055.

References

[1] C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. In *International Conference on Management of Data (SIGMOD)*, 2001. 5

[2] S. Ali, N. Sulaiman, A. Mustapha, and N. Mustapha. K-Means clustering to improve the accuracy of decision tree response classification. *Information Technology Journal*, 2009. 2, 7

[3] F. R. Bach and Z. Harchaoui. Diffrac: a discriminative and flexible framework for clustering. In *Advances in Neural Information Processing Systems (NIPS)*, 2008. 7

[4] H. R. Bonab and F. Can. GOOWE: Geometrically optimum and online-weighted ensemble classifier for evolving data streams. *Transactions on Knowledge Discovery from Data (TKDD)*, 2018. 1

[5] T. E. Boult, S. Cruz, A. R. Dhamija, M. Günther, J. Henrydoss, and W. J. Scheirer. Learning and the unknown: Surveying steps toward open world recognition. In *AAAI Conference on Artificial Intelligence*, 2019. 2

[6] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision (ECCV)*. 1, 2, 7

[7] M. Çelik, F. Dadaşer-Çelik, and A. Ş. Dokuz. Anomaly detection in temperature data using DBSCAN algorithm. In *International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, 2011. 2

[8] S. Chakraborty, N. Nagwani, and L. Dey. Performance comparison of incremental K-Means and incremental DBSCAN algorithms. *arXiv preprint arXiv:1406.4751*, 2014. 2

[9] S. Chakraborty and N. K. Nagwani. Analysis and study of incremental DBSCAN clustering algorithm. *arXiv preprint arXiv:1406.4754*, 2014. 2

[10] A. Coates and A. Y. Ng. Learning feature representations with K-Means. In *Neural networks: Tricks of the trade*. Springer, 2012. 7

[11] E. R. de Faria, A. C. P. de Leon Ferreira, J. Gama, et al. MINAS: Multiclass learning algorithm for novelty detection in data streams. *Data Mining and Knowledge Discovery*, 2016. 1

[12] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang. Efficient kNN classification algorithm for big data. *Neurocomputing*, 2016. 1, 2

[13] J. Erman, M. Arlitt, and A. Mahanti. Traffic classification using clustering algorithms. In *SIGCOMM Workshop on Mining Network Data*, 2006. 2

[14] G. Fei, S. Wang, and B. Liu. Learning cumulatively to become more knowledgeable. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016. 1

[15] M. Günther, S. Cruz, E. M. Rudd, and T. E. Boult. Toward open-set face recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017. 2

[16] U. Hahn and K. Schnattinger. Deep knowledge discovery from natural language texts. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 1997. 1

[17] J. Henrydoss. *Open-Set Intrusion Recognition Using Extreme Value Machine*. PhD thesis, University of Colorado Colorado Springs, 2019. 7

[18] J. Henrydoss, S. Cruz, E. M. Rudd, M. Günther, and T. E. Boult. Incremental open set intrusion recognition using extreme value machine. In *International Conference on Machine Learning and Applications (ICMLA)*, 2017. 4, 6

[19] L. P. Jain, W. J. Scheirer, and T. E. Boult. Multi-class open set recognition using probability of inclusion. In *European Conference on Computer Vision (ECCV)*, 2014. 2

[20] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 7

[21] B. Keith and C. Meneses. Barycentric coordinates for ordinal sentiment classification. In *Conference on Knowledge Discovery and Data Mining (KDD)*, 2017. 1

[22] K. M. Kumar and A. R. M. Reddy. A fast DBSCAN clustering algorithm by accelerating neighbor searching using groups method. *Pattern Recognition*, 2016. 2

[23] A. Li, L. Gu, and K. Xu. Fast anomaly detection for large data centers. In *Global Telecommunications Conference (GLOBECOM)*, 2010. 5

[24] A. Lourenço, H. Silva, C. Carreiras, et al. Outlier detection in non-intrusive ECG biometric system. In *International Conference on Image Analysis and Recognition (ICIAR)*, 2013. 2

[25] L. McInnes and J. Healy. Accelerated hierarchical density based clustering. In *International Conference on Data Mining (ICDM) Workshops*. IEEE, 2017. 4, 5

[26] L. McInnes, J. Healy, and S. Astels. HDBSCAN: Hierarchical density based clustering. *Journal of Open Source Software (JOSS)*, 2017. 4

[27] P. Paliwal and M. Sharma. Enhanced DBSCAN outlier detection. *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*, 2013. 2

[28] D. Ravichandran, P. Pantel, and E. Hovy. The terascale challenge. In *KDD Workshop on Mining for and from the Semantic Web*, 2004. 1

[29] E. M. Rudd, L. P. Jain, W. J. Scheirer, and T. E. Boult. The extreme value machine. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017. 1, 2, 3, 4, 6

[30] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult. Toward open set recognition. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2013. 1, 2, 3

[31] J. Shen, X. Hao, Z. Liang, Y. Liu, W. Wang, and L. Shao. Real-time superpixel segmentation by DBSCAN clustering algorithm. *Transactions on Image Processing (TIP)*, 2016. 2

[32] J. Shi and J. Malik. Normalized cuts and image segmentation. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2000. 7

[33] M. Singhal and S. Shukla. Centroid selection in kernel extreme learning machine using K-Means. In *International Conference on Signal Processing and Integrated Networks (SPIN)*, 2018. 1

[34] M. Syafrudin, G. Alfian, N. L. Fitriyani, and J. Rhee. Performance analysis of IoT-based sensor, big data processing, and machine learning model for real-time monitoring system in automotive manufacturing. *Sensors*, 2018. 5

[35] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. A detailed analysis of the KDD CUP 99 data set. In *Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 2009. 6

[36] M. Theodorakis, A. Vlachos, and T. Z. Kalamboukis. Using hierarchical clustering to enhance classification accuracy. In *Hellenic Conference in Artificial Intelligence*, 2004. 2, 7

[37] V. Vapnik. *Statistical learning theory*. Wiley, 1998. 5

[38] P. Viswanath and V. S. Babu. Rough-DBSCAN: A fast hybrid density based clustering method for large data sets. *Pattern Recognition Letters*, 2009. 2

[39] J. Yang, C. Ning, C. Deb, F. Zhang, D. Cheong, S. E. Lee, C. Sekhar, and K. W. Tham. k-Shape clustering algorithm for building energy usage patterns analysis and forecasting model accuracy improvement. *Energy and Buildings*, 2017. 1, 7

[40] A. Zhou, S. Zhou, J. Cao, Y. Fan, and Y. Hu. Approaches for scaling DBSCAN algorithm to large spatial databases. *Journal of Computer Science and Technology (JCST)*, 2000. 2

[41] Z.-H. Zhou. Large margin distribution learning. In *Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR)*, 2014. 2

[1] The source code is available at: https://github.com/Vastlab/C-EVM